

SENTENCE PROCESSING AS ABDUCTION+DEDUCTION

Shravan Vasishth and Geert-Jan M. Kruijff

Abstract

A sentence processing model is presented, based on abductive and deductive inference. We show that the model makes correct predictions for an array of data involving Dutch, German, Japanese, and Hindi center-embedding constructions. It has comparable or better empirical coverage with respect to several other theories of sentence processing, and can be integrated into an existing wide-coverage model, Lewis' Interference and Confusability Theory, to obtain an integrated theory of working memory constraints on human language processing.

1 Introduction

A well-known fact about English (Chomsky & Miller 1963) is that center-embedded constructions (CECs) like (1a) are more difficult for humans to process than right-embedded constructions like (1b).

- (1) a. The salmon [that the man [that the dog chased] smoked] fell off the grill.
- b. The dog chased the man [that smoked the salmon [that fell off the grill]].

Such CECs occur in several languages, such as Dutch, German, Japanese, and Hindi, as the following examples demonstrate.

- (2) a. (dat) Aad Jantje de lerares de knikkers liet helpen opruimen
that Aad Jantje the teacher the marbles let help collect
'(that) Aad let Jantje help the teacher collect the marbles.' (Kaan & Vasić 2000)
- b. (dass) die Männer haben Hans die Pferde füttern lehren
that the men have Hans the horses feed teach
'(that) the men have taught Hans to feed the horses.' (Bach *et al.* 1986)
- c. Keiko-ga Tadashi-ga Kenji-o kiraida-to omotteiru
Keiko-nom Tadashi-nom Kenji-acc hates-comp thinks
'Keiko thinks that Tadashi hates Kenji.' (Uehara & Bradley 1996)
- d. Siitaa-ne Hari-ko kitaab khariid-ne-ko kahaa
Sita-erg Hari-dat book buy-inf-acc said
'Sita told Hari to buy a/the book.' (Vasishth 2001)

Several experimental studies have investigated Dutch, German, Japanese, and Hindi (see, for example, (Bach *et al.* 1986), (Kaan & Vasić 2000), (Lewis 1998), (Babyonyshev & Gibson 1999), (Uehara & Bradley 1996), and (Vasishth 2001)), and as a result we now have a body of interesting, empirically determined facts about relative difficulties in processing these kind of sentences, and reading time differences during real-time processing.

Two theories that address the question of a cross-linguistically robust account of CEC processing are: Gibson's (Gibson 1998) Syntactic Prediction Locality Theory (SPLT), based on integration cost and memory cost; and Lewis' (Lewis 1998) Interference and Confusability Theory (hereafter, ICT), which relies on constraints on working memory during comprehension.¹ These theories make correct predictions for several languages, but are unable to account for all the processing difficulties in Hindi CECs. This is discussed in detail in (Vasishth 2001) (this volume), which showed that (i) although the ICT can account for processing difficulty of verbs, it is unable to account for differences in processing nouns; and (ii) the SPLT, whose complexity metric relies on the number of discourse referents introduced in a sentence, cannot account for some key Hindi processing facts.

Lewis' ICT focuses on processing difficulty at verbs, and makes the correct predictions for the processing of verbs in all the languages under consideration; however, it is less clear what the predictions are of the ICT for processing difficulty at nouns. One possibility is to add the metric proposed in this paper to the ICT; this has the advantage of maintaining the wide coverage of the ICT and of extending it to account for the Hindi data. We propose the abductive-inference based model as such an addition to the ICT.

¹This is by no means an exhaustive list of theories relating to sentence processing— we choose to discuss these two theories because they have wide empirical coverage for the questions we address here.

The structure of the paper is as follows. Section 2 outlines the main proposal: an algorithm, a complexity metric, and the relationship between the two; Section 3 illustrates the operation of the model by giving several derivations for the Dutch, German, Japanese, and Hindi facts; and Section 4 concludes the paper.

2 Processing as abduction+deduction: The main proposal

The processing model we propose for explaining the complexity of center-embedded constructions is based on a combination of *abduction* and *deduction*.²

The basic idea is as follows. We assume that we have a grammar, \mathcal{G} , for a particular natural language, \mathcal{L} . \mathcal{G} defines what types of functional categories (or predicate-valency frames) we can encounter in \mathcal{L} , and how these functions can combine with their arguments. Throughout this paper we will assume that \mathcal{G} is a categorial grammar, and that the basic types of functional categories can be derived directly from \mathcal{G} 's lexicon.

We can extract the list of types of functional categories from \mathcal{G} 's lexicon. Disregarding the specific words that each of these functional categories have been assigned to in the lexicon, we can consider this list essentially as providing us with schemas elucidating how words (of particular categories) can be combined. For example, the intransitive verbs give us the schema $f(NP)$, the transitive verbs $f(NP1, NP2)$, and so on. We regard this list as our collection of *hypothesis formers*, \mathcal{H} . We employ \mathcal{H} in the following way.

When we process a sentence, we do so by starting at the beginning of the sentence, and proceeding word by word towards the end of the sentence.³ In center embeddings, we encounter NPs before we see a verb. These NPs are arguments for one or more verbs. The NPs that we have encountered at a given point during real-time processing result in unconscious abductive inferences about the possible completion of the sentence (i.e., about the kind of schema or schemas that will apply). The model relies on the assumption that a greater number of abductive inferences will result in increased processing difficulty due to an processing overload on human working memory.⁴

Put differently, whenever we encounter a word that is believed to be an argument of an as yet unseen verb (function), we assume a *hypothetical* function that would *explain* the occurrence of that word as a (projected) argument. For example, if we encounter a noun in nominative case before we have encountered a verb, we hypothesize a verbal category that would take a noun in nominative case as its argument. It is our list \mathcal{H} that provides us with

²Abduction has been discussed in the context of semantic interpretation in previous work; see, e.g., Jerry R. Hobbs & Martin (1993), Strigin (1998). Note that the abductive-inference theory we present is intended to be a model of human cognitive processes, not a practical, real-life parser for natural language applications. Perhaps our model can be extended for such applications, but our goals here are different.

³Computers do not necessarily have to do so - for example when using head-corner parsing algorithms.

⁴We assume that working memory, or short-term memory, is "...a short-duration system in which small amounts of information are simultaneously stored and manipulated in the service of accomplishing a task" (Caplan & Waters 1999).

the possible hypothesis-candidates, since \mathcal{H} includes all the (basic) types of functions that we can conceivably encounter, given \mathcal{G} .

Subsequently, whenever the parser encounters a verb, it tries to match a hypothesized function or functions against the actual functional category of the verb. If there is a match, or if the verbal category subsumes the hypothesized function,⁵ then we can instantiate the hypothesis as the encountered verbal category, and compose the verb with the noun as its argument.⁶

Abduction, then, is understood here as the kind of unconscious and instantaneous reasoning we use to advance a hypothetical function as the best explanation for the occurrence of an argument, acting on the assumption that we are trying to process a grammatical sentence. Deduction is used in the Categorial Grammar sense, as the means to subsequently try to *compose* an actually encountered function and any available, suitable argument(s). The account of processing *complexity* arises from the number of hypotheses currently active, and how difficult it is to match them against the functional categories of observed words.

In the next subsections we discuss the notion of abduction in some more detail; then we present the algorithm and the complexity metric, and the relationship between them.

2.1 Abduction

The contemporary understanding of abduction, as a third form of logical reasoning next to deduction and induction (cf. (Josephson & Josephson 1996)), is usually traced back to its discussion by the American logician, Charles S. Peirce (Kruijff 1995; Kruijff 1998). Peirce defined abduction as the following kind of inference:

*A surprising phenomenon O is observed;
but if H were to be the case, then O would follow as a matter of course.
Therefore, there is reason to believe that H .*

Thus, whereas (intuitively speaking) deduction derives a consequence from given axioms, and induction establishes a rule or generalization, abduction proposes an *explanation* for a *surprising* observation.

The surprise is the key, here. Being surprised means that either (a) we did not expect to observe (anything like) O at all, or (b) we did expect to make some observation, but it was not O . On (a), our knowledge is incomplete, whereas on (b), our knowledge is in some way incorrect. Either way, we do not at that time have sufficient knowledge to create

⁵(by “subsumes” we mean, “is more inclusive than but not inconsistent with”; see (Shieber 1986:14-16) for a precise definition.

⁶By the term “compose” we simply mean putting a function and its argument together.

a hypothesis explaining *O*- if we knew all along that *O* would happen, then why are we surprised?⁷ Peirce's claim was that only through abduction could we obtain genuinely new knowledge (on the assumption we would let ourselves be surprised, and acknowledge that fact).

Here, as in artificial intelligence research in general, we take in a substantially weaker (but more workable) position. We assume that we have at our disposal all hypothesis formers that could be possibly abduced. For our application, the list is assumed to be the smallest one given a grammar/lexicon - the set \mathcal{H} discussed above. We assume that \mathcal{H} is finite and closed.⁸ These are all reasonable assumptions since \mathcal{H} cannot have any redundant hypothesis formers (having been created from the grammar rules), and the list of schemas extracted from the grammar will be finite (if this were not so, the set of grammar rules would have to be infinite).

\mathcal{H} is created on the basis of a compilation of the lexicon (by compilation we do not mean the creation of the lexicon; rather, given a lexicon, a set of procedures, described in detail in (Kruijff 1999), are applied to it in order to compile out information present in the lexicon). In a lexicalist approach like Categorical Grammar, the lexicon determines how words can be put together. Structural rules, like the those in Multi-Modal Logical Grammar (MMLG) (Moortgat 1997), only vary the order in which words occur grammatically.⁹

The compilation of the lexicon is based on a procedure proposed for linear logic in (Hepple 1998), and extended in (Kruijff 1999) to cover a larger range of multiplicative resource logics used in MMLG. Originally, compilation was proposed for the purposes of efficient chart parsing with Lambek-style grammars, in order to overcome problems with earlier approaches (e.g., (Hepple 1992) and (König 1990)). The result of the procedure is a set of first-order functions to represent categories (i.e., there are no higher-order formulas).

Once we have a compiled version of the lexicon, we abstract away from individual words, and retain the different functional categories that are defined. Taken together, these functional categories make up \mathcal{H} . The list of hypothesis formers \mathcal{H} is assumed to be partially ordered by a simplicity criterion: simpler structures appear before the more complex ones. Examples of the simplicity criterion: monoclausal structures are simpler than biclausal ones, the intransitive-verb based hypothesis is simpler than the ditransitive verb-based hypothesis. This assumption is not arbitrary; it is based on experimental evidence from (Yamashita 1997), which showed that (Japanese) subjects prefer to complete sentences with verbs that are simpler (i.e., verbs that result in monoclausal structures) rather than more complex ones. We take this result to indicate that simpler structures are more

⁷For that reason, Peirce advanced the idea that a hypothesis is created by a "guessing instinct" because we cannot rely on reasoning from our knowledge as such. In this context it is perhaps interesting to note that Peirce was not alone in postulating a fundamental role for something like a "guessing instinct" in logic. Gödel took the same line— cf. (Parsons 1995), and the brief comparison between Gödel's ideas and Peirce's in (Kruijff 1997).

⁸This is not to be confused with the fact that the set of sentences is infinite.

⁹Moortgat's term for MMLG is Multimodal categorial grammar. We follow the terminology used in Kruijff (2001).

accessible than more complex ones, and model this assumption by the partial ordering (the ordering is partial because it is possible that there is no way to specify relative simplicity between a given pair of hypotheses). We leave aside the issue of precisely defining the ordering criteria for the moment.¹⁰

2.2 Some definitions

Next, we define some terms that we use in the proposed algorithm.

Abducible structure(s): An ABDUCIBLE STRUCTURE is a hypothesis based on the information available so far; no more hypotheses are selected than are justified by the information available up to a certain point (this will be made precise presently).

New information results in the replacement of previous hypotheses. Abduced *functions* f_i are part of the abducible structures that are taken from \mathcal{H} , and thus posit the presence of a word with a particular syntactic category. For example, in Japanese, if only a nominative NP (we represent this as $NP[nom]$) has appeared so far, $f_i(NP[nom])$ is a syntactic hypothesis that says: an intransitive verb f_i with the nominative NP will give a sentence.¹¹

Note that although a nominative case marked NP is in principle consistent with an infinite set of possible continuations, our model allows for the selection of only those hypotheses from the hypothesis formers \mathcal{H} that are *minimally consistent* with the nominative NP. We define minimal consistency as follows:

Minimal consistency:

There are two cases: (i) only a list of NPs has been seen so far, (ii) a list of NPs and a verb, or only a verb, has been seen so far.

- (i) If only a list of NPs has been seen so far: A list of hypotheses $H', H' \subset H$, is minimally consistent with a given list of nouns NPs iff each hypothesis $h \in H'$ is able to take the NP's as arguments without positing any new, unseen arguments.

¹⁰Lewis (personal communication) informs me that Uehara has found that Japanese subjects prefer two clauses over one in incomplete sentences beginning with two nominative-case marked NPs; in Japanese, this sequence could be continued either with a stative verb or a bi-clausal structure. The simplicity criterion given here wrongly predicts that the stative verb (monoclausal structure) is ordered before the biclausal one. It is likely that the simplicity criterion is an oversimplification, and that a more sophisticated set of decision criteria are needed (such as frequency of certain syntactic structures) in order to determine the ordering. I leave this question open for future research.

¹¹The subscript on f is merely a notational device used in the derivations in Section 3 for improving readability.

- (ii) A list of hypotheses $H', H' \subset H$, is minimally consistent with a verb, or a given list of nouns NPs and a verb, iff each hypothesis $h \in H'$ is able to take any of the seen NP's as arguments (given the valency-frame of the verb that has been seen); if the verb requires any new, unseen argument(s) and/or is an argument of another as-yet-unseen verb f_i , the unseen argument(s) and/or the function f_i are posited. Any unseen arguments that the function f_i would require are also posited.

An example illustrating the first clause above of minimal consistency is as follows. Suppose that, during the course of processing a Japanese sentence, we have seen only one nominative NP so far. In that case, a hypothesis satisfying minimal consistency is $f_i(NP[nom])$, and one *violating* minimal consistency is: $f_i(NP[nom], x)$, where x is a hypothesized, new, unseen NP. By contrast, if, after we see the first nominative NP, we see a second nominative NP, the minimally consistent hypotheses are now $f_i(NP1[nom], NP2[nom])$, where f_i is a stative verb, and $f_{i+1}(NP1[nom], f_{i+2}(NP2[nom]))$, i.e., a center-embedded structure.

The second clause of minimal consistency can be exemplified as follows. Suppose we are processing a sentence in Japanese, and we first see a verb V1 like *itta-to*, 'said[past]-complementizer'. Here, the hypothesis will be $f_i(x, V1(y, z))$; since V1 is necessarily an embedded verb (due to the presence of the complementizer), there is a function f_i (with some subject NP x) that takes a clause headed by V1 as an argument, and V1 takes as-yet-unseen arguments y and z .

There is some psychological motivation for the minimal consistency constraint. Yamashita (Yamashita 1997) has conducted Japanese sentence completion tasks where she presented subjects with incomplete sentences containing only a series of NPs which they were asked to complete. She found that subjects tended to use verbs that subcategorized only for the NPs present, but not verbs that would require adding new, unseen NPs. The first author of this paper obtained a similar result for Hindi in a pilot study.¹²

Turning next to the issue of processing verbs after the nouns have been seen, the model uses a process of matching the verb to the hypothesized function or functions, in the manner defined below.

Matching: A verb V MATCHES with a function f_i iff V has a valency that is identical with that of f_i .

An NP can MATCH with a posited NP argument iff its case marking, person, number, gender, and other information, is consistent with that of the posited argument.

With these definitions in place, we turn next to the algorithm, based on which the complexity metric is defined.

¹²Of course, it is still an open question whether completion preferences relate to parsing preferences.

2.3 The algorithm

The processing algorithm works as follows.

Init: Set the queue data structure \mathfrak{S} to \emptyset , set the scanning pointer to position 0.

Scan: Scan the next word w_i , moving the pointer to the next position.

Lookup: Lookup the scanned word w_i in the lexicon of \mathcal{G} .

Process: This is the main part of the algorithm.

```

if  $\mathfrak{S} = \emptyset$  then
  check the category  $\mathbf{C}$  of  $w_i$ :
  if  $\mathbf{C}$  is a function category then
     $\mathfrak{S} = \mathfrak{S} \cup \{\mathbf{C}\}$ 
  else
     $\mathfrak{S} = \mathbf{abduce}(\mathcal{H}, \mathbf{C}, \mathfrak{S})$ 
  end if
else  $\{\mathfrak{S} \neq \emptyset\}$ 

  if the category of  $w_i$  is a function category  $\mathbf{C}$  then
     $\mathfrak{S} = \mathbf{deduce}(\mathbf{C}, \mathfrak{S})$ 
    or (failing that)  $\mathfrak{S} = \mathfrak{S} \cup \{\mathbf{C}\}$ 
  else  $\{\mathbf{C}$  is not a function category  $\mathbf{C}\}$ 
     $\mathfrak{S} = \mathbf{deduce}(\mathbf{C}, \mathfrak{S})$ 
    or failing that  $\mathfrak{S} = \mathbf{abduce}(\mathcal{H}, \mathbf{C}, \mathfrak{S})$ 
    If the latter step fails, or we arrive at the latter step
    and  $w_i$  is the last word in the sentence, then FAIL.
  end if
end if

```

$\mathfrak{S} = \mathbf{deduce}(\mathbf{C}, \mathfrak{S})$:

Given a category \mathbf{C} and a structure \mathfrak{S} , if \mathbf{C} is an argument then try to combine it with a hypothesis or function in \mathfrak{S} , starting with the outermost hypothesis/function first (FIFO). Else, \mathbf{C} is a function, and try to match it against a hypothesis h in \mathfrak{S} , such that \mathbf{C} is either equal to h or subsumes it. Failing all that, throw an exception stating that the word with category \mathbf{C} cannot be combined with anything in the structure.

$\mathfrak{S} = \mathbf{abduce}(\mathcal{H}, \mathbf{C}, \mathfrak{S})$: Given a list of possible hypotheses \mathcal{H} , a category \mathbf{C} , and a structure \mathfrak{S} , find the minimally consistent hypothesis (or hypotheses) h in \mathcal{H} that takes \mathbf{C} as an argument, and which can be combined with \mathfrak{S} either as an argument of a hypothesis/function in \mathfrak{S} , or as a function taking the outermost hypothesis/function in \mathfrak{S} as its argument. If no such hypothesis h can be found, then FAIL. Otherwise, integrate \mathbf{C} and h into \mathfrak{S} and return the updated structure. The hypotheses abduced in this step are ordered by the simplicity criterion.

Processing starts with **Init**. Subsequently, we cycle through **Scan-Lookup-Process**, either until we FAIL or until we arrive to the end of the sentence. If the structure \mathfrak{S} contains no unmatched hypotheses, then the sentence is grammatical and can be assigned \mathfrak{S} ; otherwise, the sentence is considered ungrammatical on \mathcal{G} .

To repeat an earlier example from Japanese: two nominative case marked NPs starting a sentence could be followed either by a stative predicate (2.3a), or a nested dependency construction with a single level of embedding (2.3b).

- (3) a. $f_2(NP1[nom], NP2[nom])$
 b. $f_3(NP1[nom], f_4(NP2[nom]))$

These are two hypotheses selected from \mathcal{H} . No other hypotheses are selected because these are the only two that are minimally consistent, given the information so far. These hypotheses are based on the grammatical possibilities in Japanese, and since a single clause sentence has a simpler structure than a sentence with an embedded clause, the hypotheses are ordered as shown above. Next, the appearance of an accusative case marked NP will result in these hypotheses being discarded and the new hypothesis being selected:

- (4) $f_5(NP1[nom], f_6(NP2[nom], NP3[acc]))$

Since the number of hypotheses has fallen from two to one, the model predicts faster processing at the accusative NP. This prediction is borne out, as discussed further on. We turn next to the complexity metric.

2.3.1 The complexity metric

The complexity metric has two components: **ABDUCTION COST**, the cost associated with the abductive process, and **MISMATCH COST**, the cost associated with a mismatch between an encountered verb and abducted functions.

Abduction cost: This reflects the increasing processing load as sentence fragments appear incrementally. The abduction cost is the sum of the number of NPs seen so far, the number of functions f_i that are posited, and the total number of distinct hypotheses abducted at a given point. These three sub-components are intended to reflect the load in working memory of: (a) storing an increasing number of NPs; (b) positing functions; and (c) storing hypotheses.

Mismatch cost: We assume that the (queued) hypotheses are unanalyzable units at first. By the term “unanalyzable units” we simply mean that when a hypothesis like $f_i(NP1, f_j(NP2))$ is present in working memory and a verb is encountered, any attempt to match the verb with any of the functions f present in the hypothesis must be a left to right

depth first search; the verb cannot directly match the right function. During this search process, every time a verb fails to match with a hypothesized function, there is a mismatch cost of one.

The numerical value associated with each sub-component in the metric is assumed to be 1 and the components are assumed to be additive. This is merely a convenience, and nothing crucial hinges on this assumption. In a fully implemented version of this model, the unit costs associated with each component will be associated with precise reading time predictions.

The complexity metric applies in conjunction with the application of the algorithm: at each stage when the algorithm incrementally builds/revises the list of possible hypotheses, the complexity metric is used to compute the processing cost at that point.

In the next section, we provide some illustrations of the empirical coverage of this processing model.

3 The empirical coverage

3.1 Japanese

Note that in the following discussion, the verbs in nested sentences are numbered in reverse order of occurrence, i.e., the matrix verb, which appears last, is V1. The numbers do not reflect the verbs' valencies; this reverse numbering convention is merely in order to highlight the difference from Dutch (discussed later).

3.1.1 Gibson's (1998) data

Gibson (Gibson 1998) has shown that (5a) is less acceptable than (5b).

- (5) a. obasan-ga bebiisitaa-ga ani-ga imooto-o izimeta-to itta-to
 aunt-nom babysitter-nom brother-nom sister-acc teased-comp. said-comp.
 omotteiru
 thinks
 'The aunt thinks that the babysitter said that the elder brother teased the younger sister.'
- b. bebiisitaa-ga ani-ga imooto-o izimeta-to itta-to obasan-ga
 babysttr.-nom brother-nom sister-acc teased-comp. said-comp. aunt-nom
 omotteiru
 thinks
 'The aunt thinks that the babysitter said that the elder brother teased the younger

sister.'

First, consider the application of the algorithm for (5a):

Step 1:

Input	Abduction/deduction	Abduction Cost	mismatch cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0

Here, given only the first NP (*obasan-ga*), a sentence with an intransitive verb (IV), denoted by f_1 , is abducted. This contributes 3 to our cost so far (abduction cost, composed of the number of NPs seen so far (1), plus the number of functions abducted (1), plus the number of hypotheses abducted (1); mismatch cost is currently 0).

Step 2:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		

Given the second NP (*bebisitaa-ga*), and given that both the NPs seen so far are nominative case marked, the abducible structures are: a stative predicate taking two nominative arguments ($f_2(\text{NP1}, \text{NP2})$), and a center embedded construction ($f_3(\text{NP1}, f_4(\text{NP2}))$). The abduction cost here is 7: 2 NPs, 3 functions, and 2 hypotheses.

Step 3:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		
NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
	$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		

We now have three nominative NPs, and so we either have an embedded stative predicate, as in $f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$, or a center embedding, as in $f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$. The abduction cost is now 10.

Step 4:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		
NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
	$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		
NP4-o	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4})))$	$4+3+1=8$	0

$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4})))$ is abducted because the fourth NP is marked with accusative case, and so there must be at least one embedding with a transitive embedded verb. The abduction cost is now 8; i.e., the model predicts that processing will take less time at this fourth NP, compared to the third NP.

Step 5:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		
NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
	$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		
NP4-o	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4})))$	$4+3+1=8$	0
V3	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4})))$	$4+2+1=7$	2

Here, the next word is *izimeta-to*, ‘teased-complementizer’, and a deduction is performed in the following manner:

(i). V3 tries to match f_{10} in

$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4}))) \Rightarrow \text{failure.}$

This matching attempt fails because the outermost function f_{10} has a valency frame that doesn’t match the actual verb’s.

(ii). V3 tries to match f_{11} in

$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4}))) \Rightarrow \text{failure.}$

Here, again, the failure occurs due to the valency frame of the verb not matching that of the next function.

(iii). V3 tries to match f_{12} in

$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4}))) \Rightarrow f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4})))$

This succeeds because the valency frame of the verb matches that of the next function. The cost now is the sum of the abduction cost (7) plus the number of failed matches (2): 9. Notice that the number of abducted functions is now 2, not 3; this is because one of the abducted functions has already been resolved by its matching with V3.

Step 6:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		
NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
	$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		
NP4-o	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4})))$	$4+3+1=8$	0
V3	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4})))$	$4+2+1=7$	2
V2	$f_{10}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4})))$	$4+1+1=6$	1

The deductive process goes as follows:

(i). V2 tries to match f_{10} in

$$f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4}))) \Rightarrow \text{failure.}$$

(ii). V2 tries to match f_{11} in

$$f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4}))) \Rightarrow f_{10}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4})))$$

V2 fails to match f_{10} , but successfully matches f_{11} . The cost is now 7 (the abduction cost, 6, plus the mismatch cost, 1).

Step 7:

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
	$f_3(\text{NP1}, f_4(\text{NP2}))$		
NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
	$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		
NP4-o	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, f_{12}(\text{NP3}, \text{NP4})))$	$4+3+1=8$	0
V3	$f_{10}(\text{NP1}, f_{11}(\text{NP2}, \text{V3}(\text{N3}, \text{NP4})))$	$4+2+1=7$	2
V2	$f_{10}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4})))$	$4+1+1=6$	1
V1	$\text{V1}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4})))$	$4+0+0=4$	0

The deduction in this case is immediate:

V1 tries to match f_{10} in

$$f_{10}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4}))) \Rightarrow \text{V1}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3}, \text{NP4})))$$

Here, V1 matches the outermost abducted function f_{10} immediately, and the parse is completed. The cost at this stage is 4.

The total cost (the sum of the costs at each step) gives us the overall complexity of the sentence relative to other sentences. So, in this case, the total cost is 48.

By contrast, (5b)’s processing yields a lower total cost of 38:

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2-ga	$f_2(\text{NP1}, \text{NP2})$ $f_3(\text{NP1}, f_4(\text{NP2}))$	$2+3+2=7$	0
3	NP3-o	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+2+1=6$	0
4	V3	$f_5(\text{NP1}, \text{V3}(\text{NP2}, \text{NP3}))$	$3+1+1=5$	1
5	V2	$f_7(x, \text{V2}(\text{NP1}, \text{V3}(\text{NP2}, \text{NP3})))$	$4+1+1=6$	0
6	NP4-ga	$f_7(\text{NP4}, \text{V2}(\text{NP1}, \text{V3}(\text{NP2}, \text{NP3})))$	$4+1+1=6$	0
7	V1	$\text{V1}(\text{NP4}, \text{V2}(\text{NP1}, \text{V3}(\text{NP2}, \text{NP3})))$	$4+0+1=5$	0

Table 1: (5b)

Note that in Step 5 above, the appearance of an embedded verb results in an abducted hypothesis involving a matrix verb and a nominal argument. This is because V2 has the complementizer *-to*, which requires it to be an embedded verb; i.e., the second clause in the definition of minimal consistency applies.

3.1.2 Nakatani et al. (2000)

(Nakatani *et al.* 2000) conducted several off-line acceptability rating questionnaire experiments with Japanese; their results may be summarized as follows:¹³

Nakatani et al. found that double embeddings are less acceptable than left branching structures. The examples below illustrate the relevant structures.

- (6) a. [obasan-wa [bebiisitaa-ga [imooto-ga naita-to] itta-to] omotteiru]
 aunt-top babysitter-nom sister-nom cried-comp. said-comp. thinks
 ‘The aunt thinks that the babysitter said that the younger sister cried.’

¹³Note: the English glosses are sometimes different from (Nakatani *et al.* 2000).

- b. [imooto-ga naita-to] bebiisita-ga itta-to] obasan-wa omotteiru]
 sister-nom cried-comp. babysitter-nom said-comp. aunt-top thinks
 ‘The aunt thinks that the babysitter said that the elder brother teased the younger sister.’

Our model makes the correct prediction about this set of examples, as the following two derivations show.

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-wa	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2-ga	$f_2(\text{NP1}, \text{NP2})$	$2+3+2=7$	0
		$f_3(\text{NP1}, f_4(\text{NP2}))$		
3	NP3-ga	$f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+5+2=10$	0
		$f_7(\text{NP1}, f_8(\text{NP2}, f_9(\text{NP3})))$		
4	V3-to	$f_7(\text{NP1}, f_8(\text{NP2}, \text{V3}(\text{NP3})))$	$3+2+1=6$	2
5	V2-to	$f_7(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3})))$	$3+1+1=5$	1
6	V1	$\text{V1}(\text{NP1}, \text{V2}(\text{NP2}, \text{V3}(\text{NP3})))$	$3+0+1=4$	0

Table 2: Double nesting, total cost is 40 for (6a)¹⁴

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
2	V3-to	$f_2(\text{V3}(\text{NP1}), x)$	$2+1+1=4$	0
3	NP2-ga	$f_2(\text{V3}(\text{NP1}), \text{NP2})$	$2+1+1=4$	0
4	V2-to	$f_3(\text{V2}(\text{V3}(\text{NP1}), \text{NP2}), y)$	$3+1+1=5$	0
5	NP3-ga	$f_3(\text{V2}(\text{V3}(\text{NP1}), \text{NP2}), \text{NP3})$	$3+1+1=5$	0
6	V1	$\text{V1}(\text{V2}(\text{V3}(\text{NP1}), \text{NP2}), \text{NP3})$	$3+0+1=4$	0

Table 3: Left branching, total cost is 25 for (6b)

Moreover, Nakatani et al. found that in double embeddings intransitive V3’s are more acceptable than transitive V3’s. Examples of these structures are shown below.

- (7) a. haha-ga titi-ga fukigen-na akatyan-ga naita-to itta-to
 mother-nom father-nom fussy baby-nom cried-comp. said-comp.
 omotteiru
 thinks
 ‘My mother thinks that my father said that the fussy baby cried.’

¹⁴In examples like (6a), we predict a fall in reading time at V3 due to a hypothesis being eliminated. We do not have any data yet to confirm or disconfirm this prediction.

- b. obasan-ga syoojiki-na bebisitaa-ga ani-ga imooto-o izimeta-to
 aunt-nom honest babysitter-nom brother-nom sister-acc teased-comp.
 itta-to omotteiru
 said-comp. thinks
 ‘My aunt thinks that the honest babysitter said that my brother teased my sister.’

The model makes the correct prediction since (7)a has cost 40 and (7)b has cost 48. See earlier derivations (Table 2 and the full derivation for (5a)) respectively).

3.1.3 Yamashita (1997)

Yamashita (Yamashita 1997) investigated the effect of word order and case marking on the processing of Japanese. One of her experiments is a moving window task involving three conditions:

Condition A. Canonical order, with 4NPs and 2 verbs:

[NP1-nom NP2-dat [NP3-nom NP4-acc V2] V1]

Condition B. Same structure as in Condition A, but scrambled NP3 and NP4:

[NP1-nom NP2-dat [NP4-acc NP3-nom V2] V1]

Condition C. Same structure as in Condition A, but scrambled NP1, NP2, NP3 and NP4:

[NP2-dat NP1-nom [NP4-acc NP3-nom V2] V1]

The results for Condition A are interesting in the context of the present model;¹⁵ consider the example below.

- (8) [denwa-de hansamu-na gakusei-ga sensei-ni [tumetai koibito-ga nagai
 phone-on handsome student-nom teacher-dat cold girlfriend-nom long
 tegami-o yabutta-to] itta]
 letter-acc tore-comp. said
 ‘On the phone, a handsome student told the teacher that the cold-hearted girlfriend
 had torn up the letter.’

Yamashita found that reading times rose steadily in such examples till the accusative marked NP, and then fell at the accusative NP.

¹⁵In this paper, we do not discuss the effect of word order variation since this introduces issues of pragmatics that the model currently does not take into account. The model can, however, be extended to incorporate constraints from pragmatics; essentially, the idea would be to include information from the pragmatics of an utterance in the abductive process.

The present model predicts this pattern, as shown below.

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-ga	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2-ni	$f_2(\text{NP1}, \text{NP2})$	$2+1+1=4$	0
3	NP3-ga	$f_3(\text{NP1}, \text{NP2}, f_4(\text{NP3}))$ $f_5(\text{NP1}, f_6(\text{NP2}, \text{NP3}))$	$3+4+2=9$	0
4	NP4-o	$f_7(\text{NP1}, \text{NP2}, f_8(\text{NP3}, \text{NP4}))$	$4+2+1=7$	0
5	V2	$f_7(\text{NP1}, \text{NP2}, \text{V2}(\text{NP3}, \text{NP4}))$	$4+1+1=6$	1
6	V1	$\text{V1}(\text{NP1}, \text{NP2}, \text{V2}(\text{NP3}, \text{NP4}))$	$4+0+0=4$	0

Table 5: (3.1.3)

Before step 4, the reading time is predicted to rise steadily. At step 4, a fall in reading time is predicted since the number of hypotheses falls from two to one, and the number of functions is now one.

3.2 Dutch and German

3.2.1 Dutch: Kaan et al. (2000)

Turning next to Dutch, Kaan and Vasić (Kaan & Vasić 2000) conducted several moving window studies and found the following.

Fact 1: Double embeddings harder than single embeddings

Examples of each type are shown below:

- (9) a. De leider heeft Paul Sonya het kompas helpen leren gebruiken tijdens de
the leader has Paul Sonya the compass help teach use during the
bergtocht
hike
'The leader helped Paul teach Sonya to use the compass during the hike.'
- b. Met aanwijzingen van de leider heeft Paul Sonya het kompas helpen
with directions of the leader has Paul Sonya the compass teach
gebruiken tijdens de bergtocht
use during the hike
'With the leader's directions Paul taught Sonya to use the compass during the hike.'

Double embeddings have a cost of 50:

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2	$f_2(\text{NP1}, \text{NP2}), f_3(\text{NP1}, f_4(\text{NP2}))$	$2+3+2=7$	0
3	NP3	$f_5(\text{NP1}, \text{NP2}, \text{NP3}))$ $f_6(\text{NP1}, f_7(\text{NP2}, \text{NP3}))$ $f_8(\text{NP1}, f_9(\text{NP2}, f_{10}(\text{NP3})))$	$3+6+3=12$	0
4	NP4	$f_{11}(\text{NP1}, f_{12}(\text{NP2}, \text{NP3}, \text{NP4}))$ $f_{13}(\text{NP1}, f_{14}(\text{NP2}, f_{15}(\text{NP3}, \text{NP4})))$	$4+5+2=11$	0
5	V1	$V1(\text{NP1}, f_{14}(\text{NP2}, f_{15}(\text{NP3}, \text{NP4})))$	$4+2+1=7$	0
6	V2	$V1(\text{NP1}, V2(\text{NP2}, f_{15}(\text{NP3}, \text{NP4})))$	$4+1+1=6$	0
7	V3	$V1(\text{NP1}, V2(\text{NP2}, V3(\text{NP3}, \text{NP4})))$	$4+0+0=4$	0

Table 6: total cost is 50 for (9a)

Single embeddings have a lower cost of 30.

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2	$f_2(\text{NP1}, \text{NP2}), f_3(\text{NP1}, f_4(\text{NP2}))$	$2+3+2=7$	0
3	NP3	$f_5(\text{NP1}, \text{NP2}, \text{NP3}))$ $f_6(\text{NP1}, f_7(\text{NP2}, \text{NP3}))$ $f_8(\text{NP1}, f_9(\text{NP2}, f_{10}(\text{NP3})))$	$3+6+3=12$	0
4	V1	$V1(\text{NP1}, f_7(\text{NP2}, \text{NP3}))$	$3+1+1=5$	0
5	V2	$V1(\text{NP1}, V2(\text{NP2}, \text{NP3}))$	$3+0+0=3$	0

Table 7: total cost is 30 for (9b)

Kaan and Vasić found that RTs increased with each incoming NP, and fell at the innermost verb, which is what our model predicts. In the present model, the NP reading times are predicted to rise due to the increase in the number of abducted functions, and a fall in reading time is predicted at the first verb due to the elimination of some hypotheses (see derivations above to see how exactly this happens).

3.2.2 Dutch and German: Bach et al. (1986)

Bach et al. (Bach *et al.* 1986) showed that Dutch crossed dependencies were easier to process for native Dutch speakers than German nested dependencies are for native German speakers. Examples of crossed Dutch and nested German dependencies are shown below:

- (10) a. NP1 NP2 NP3 V1 V2 V3

Jan Piet Marie zag laten zwemmen
 Jan Piet Marie saw make swim

‘Jan saw Piet make Marie swim.’

b. NP1 NP2 NP3 V3 V2 V1

...dass Hans Peter Marie schwimmen lassen sah
 ...that Hans Peter Marie swim make saw

‘...that Hans saw Peter make Marie swim.’

The Dutch CECs are called crossed because of the fact that the verbs and the subjects they link with form crossing chains (NP1 NP2 NP3 V1 V2 V3), and the German CECs are nested since the pattern is NP1 NP2 NP3 V3 V2 V1.

Our model predicts that Dutch center embeddings will be more acceptable since, as shown in Tables 6 and 7, in Dutch, there will be no mismatch cost; in the analogous German examples, however, there will be a mismatch cost associated with each embedded verb.

3.3 Hindi

Vasishth (Vasishth 2001) conducted a self-paced reading time study and found that in center embeddings, accusative case marking on direct objects in Hindi (which marks specificity in the case of inanimate objects), makes processing harder. Examples of single center embeddings are shown below.

- (11) a. Siitaa-ne Hari-ko [kitaab khariid-ne-ko] kahaa
 Sita-erg Hari-dat book buy-inf told
 ‘Sita told Hari to buy a/the book.’
 b. Siitaa-ne Hari-ko [kitaab-ko khariid-ne-ko] kahaa
 Sita-erg Hari-dat book-acc buy-inf told
 ‘Sita told Hari to buy the book.’

The model predicts that in the case of (11a), there will be only one hypothesis by the time the third NP is processed, whereas in (11b), there will be two hypotheses at the third NP. These two hypotheses arise because of the fact that both the dative and accusative case markings in Hindi are marked by the suffix/postposition *-ko*, and because Hindi has extremely free word order. The phonologically similar case marking combined with the possibility of reordering NP2 and NP3 results in two possible hypotheses.

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-ne	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2-ko	$f_2(\text{NP1}, \text{NP2})$	$2+1+1=4$	0
3	NP3	$f_3(\text{NP1}, f_4(\text{NP2}, \text{NP3}))$	$3+2+1=6$	0
4	V2	$f_2(\text{NP1}, \text{V2}(\text{NP2}, \text{NP3}))$	$3+1+1=5$	1
5	V1	$\text{V1}(\text{NP1}, \text{V2}(\text{NP2}, \text{NP3}))$	$3+0+0=3$	0

Table 8: total cost is 22 for (11a)

Step	Input	Abduction/deduction	Abduction Cost	Mismatch Cost
1	NP1-ne	$f_1(\text{NP1})$	$1+1+1=3$	0
2	NP2-ko	$f_2(\text{NP1}, \text{NP2})$	$2+1+1=4$	0
3	NP3-ko	$f_3(\text{NP1}, f_4(\text{NP2}, \text{NP3}))$ $f_5(\text{NP1}, f_6(\text{NP3}, \text{NP2}))$	$3+4+2=9$	0
4	V2	$f_2(\text{NP1}, \text{V2}(\text{NP2}, \text{NP3}))$	$3+1+1=5$	1
5	V1	$\text{V1}(\text{NP1}, \text{V2}(\text{NP2}, \text{NP3}))$	$3+0+0=3$	0

Table 9: total cost is 24 for (11b)

Similar predictions hold for double embeddings, but a discussion is omitted. For details, see (Vasishth 2001).

4 Concluding remarks

A hybrid abductive/deductive model of human language processing is proposed, based on existing psycholinguistic results. An important observation is that many of the mechanisms proposed have correlates in other theories. For example, the number of NPs seen up to a given point are counted as part of the abduction cost; this corresponds to the number of discourse referents, which is a critical component in Gibson’s model. Our contribution is to propose a very general general perceptual mechanism— abduction— as the key process that allows an incremental parse, given a particular grammar \mathcal{G} for the relevant language \mathcal{L} . The parsing mechanism in our model is very similar to the well-known shift-reduce parser and the Earley parser (Aho & Ullman 1993), (Sikkel 1997); due to space constraints, we do not present a detailed discussion of the similarity and differences. See (Vasishth in progress) for a discussion.

The model fares better than existing accounts for the data considered here. For example, none of the existing theories can currently account for the fall in reading times at the accusative verb in Japanese, and at the first verb in Dutch; and Gibson’s model (Gibson 1998) appears to make incorrect predictions for the rising reading times for verbs (see (Kaan & Vasić 2000) for details). However, it remains to be seen whether the predictions

it makes are all borne out. For example, the model predicts that there will be a fall in reading time when the number of abducted hypotheses is reduced in working memory to one as a result of new incoming information. This happens to be the correct prediction for Yamashita's Japanese data and Kaan and Vasić's Dutch data, but we do not have enough data yet to determine whether this prediction is borne out (for example) for (5b).

Further, we currently do not have a precise account for the scrambling facts (e.g., those presented in (Yamashita 1997)). One reason that we hesitate to extend our model for scrambling is that word order variation is almost always correlated with a particular discourse context, and yet studies on scrambling and processing like Yamashita's (Yamashita 1997) assume that processing of a scrambled sentence presented to subjects out of the blue (i.e., without any discourse context) can be compared with unscrambled correlates. Pilot sentence completion studies using Hindi, conducted by the first author, indicate that subjects find scrambled sentences less acceptable than unscrambled ones (these were presented without any preceding discourse context). We must therefore await further empirical work before any valid conclusions can be drawn about the processing of scrambled sentences.

There are some facts that our model fails to capture. For example, Nakatani et al. found that a singly nested, 5 NP stack was more acceptable than doubly nested, 3-4 NP stacks ((Lewis 1993) was the first to discuss such 5-NP structures). The relevant examples are given below and the derivation for (12a) is shown in Table 4.

- (12) a. tuma-wa kakarityoo-ni uranaisi-ga otto-ni seekoo-o
 wife-nom chief-clerk-dat fortune-teller-nom husband-dat success-acc
 yakusoku-sita-to ziman-sita
 promised-comp. boasted
 'The wife boasted to the chief clerk that the fortune-teller promised the husband that he'd succeed.'
- b. haha-ga titi-ga fukigen-na akatyan-ga naita-to itta-to
 mother-nom father-nom fussy baby-nom cried-comp. said-comp.
 omotteiru
 thinks
 'My mother thinks that my father said that the fussy baby cried.'
- c. obasan-ga syoojiki-na bebisitaa-ga ani-ga imooto-o izimeta-to
 aunt-nom honest babysitter-nom brother-nom sister-acc teased-comp.
 itta-to omotteiru
 said-comp. thinks
 'My aunt thinks that the honest babysitter said that my brother teased my sister.'

Input	Abduction/deduction	Abduction Cost	Mismatch Cost
NP1-wa	$f_1(\text{NP1})$	$1+1+1=3$	0
NP2-ni	$f_2(\text{NP1}, \text{NP2})$	$2+1+1=4$	0
NP3-ga	$f_3(\text{NP1}, \text{NP2}, f_4(\text{NP3}))$	$3+2+1=6$	0
NP4-ni	$f_5(\text{NP1}, \text{NP2}, f_6(\text{NP3}, \text{NP4}))$	$4+2+1$	0
NP5-o	$f_7(\text{NP1}, \text{NP2}, f_8(\text{NP3}, \text{NP4}, \text{NP5}))$	$5+2+1=8$	0
V2-to	$f_7(\text{NP1}, \text{NP2}, \text{V2}(\text{NP3}, \text{NP4}, \text{NP5}))$	$5+1+1=7$	1
V1	$\text{V1}(\text{NP1}, \text{NP2}, \text{V2}(\text{NP3}, \text{NP4}, \text{NP5}))$	$5+0+0$	0

Table 10: total cost is 41 for (12a)

Our model incorrectly predicts that (12a) will be less acceptable than (12b) (which has cost 40) (see Table 2), but correctly predicts that it will be more acceptable than (12c), which has cost 48 (see the first derivation presented in this paper). However, we consider our model to be primarily a theory of the encoding processes that occur during NP processing, and we propose to integrate this theory of encoding-via-abductive-inference with Lewis' Interference and Confusability Theory (ICT) which is a theory of the integration of encoded NPs with verbs. Integrating the present theory with Lewis' ICT gives us a complete account of encoding and retrieval processes during sentence processing; this integrated account, we argue, makes more correct predictions than other current sentence processing models. See (Vasishth 2001) (this volume) for details.

Finally, in relation to other, similar accounts, we contend that our account is a useful generalization over accounts like the ones based on pushdown automata (Joshi 1990), or incremental processing by prediction of minimum valency as proposed in work by Scheepers *et al.* (Scheepers *et al.* 1999). Implicit in all these treatments is the idea of abductive inference. Our proposal foregrounds abduction, and demonstrates the considerable predictive power such foregrounding makes available to us: for example, thinking about processing as abduction helped us identify the components of our complexity metric. In this sense, our model is less a challenge to existing accounts than a reformulation of these in more general (although very precise) terms. Future work will consist of building a computational implementation of the integrated ICT/abductive inference model.

Acknowledgments

Thanks to Richard Lewis and John Josephson for their detailed feedback, and to Chris Brew, Takao Gunji, Martin Jansche, Keith Johnson, Neal Johnson, Brian Joseph, Mineharu Nakayama, Shari Speer, and Mark Steedman, for many useful comments. At the time of writing, the second author was a visitor at the Institute of Communicating and Collaborative Systems (ICCS), Division of Informatics, University of Edinburgh, and would therefore like to acknowledge here Mark Steedman's and Bonnie Webber's hospitality. An earlier version of this paper was presented in Japan at the Japanese Cognitive Science Meeting, 2000, at the Kobe Shoin Graduate School, and at the ESSLLI 2000 Linguistic Theory and Grammar Implementation Workshop, Birmingham, UK; we thank the audiences there for their comments. The authors alone are responsible for any errors. Please note that this is

work in progress; a more articulated version of this paper can be obtained from the authors, who can be contacted at the addresses given below. An Edinburgh Prolog implementation of the model presented here is available from the first author.

Shravan Vasishth	Dr.ir. Geert-Jan M. Kruijff
Department of Linguistics	Computational Linguistics
Ohio State University	University of the Saarland
222 Oxley Hall, 1712 Neil Avenue	Postfach 15 11 50
Columbus, OH 43210-1022 (USA)	D-66041 Saarbrücken (Germany)
Email: vasishth@ling.onio-state.edu	Email: gj@coli.uni-sb.de, gj@acm.org

References

- AHO, A. V., & J. D. ULLMAN (eds.) 1993. *Principles of Compiler Design*. New Delhi, India: Narosa Publishing House.
- BABYONYSHEV, MARIA, & EDWARD GIBSON. 1999. The complexity of nested structures in Japanese. *Language* 75(3).423–450.
- BACH, E., C. BROWN, & W. MARSLEN-WILSON. 1986. Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes* 1(4).249–262.
- CAPLAN, D., & G. S. WATERS. 1999. Verbal working memory and sentence comprehension. *Behavioral and Brain Sciences* 22.77–94.
- CHOMSKY, NOAM, & GEORGE MILLER. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, ed. by R. Duncan Luce, Robert R. Bush, & Eugene Galanter, 269–321. John Wiley.
- GIBSON, EDWARD. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition* 68.1–76.
- HEPPLE, MARK. 1992. Chart parsing lambek grammars: Modal extensions and incrementality. In *Proc. COLING 1992*.
- . 1998. Memoisation for glue language deduction and categorial parsing. In *Proceedings of the COLING-ACL Joint Conference, Montreal Canada, August. Ninth International Congress on Linguistics*. ACL.
- JERRY R. HOBBS, MARK E. STICKEL, DOUGLAS E. APPELT, & PAUL MARTIN. 1993. Interpretation as Abduction. *Artificial Intelligence* 63.69–142.
- JOSEPHSON, JOHN R., & SUSAN G. JOSEPHSON. 1996. *Abductive Inference: Computation, Philosophy, Technology*. Cambridge, UK: Cambridge University Press.

- JOSHI, ARAVIND K. 1990. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes* 5(1).1–27.
- KAAN, EDITH, & NADA VASIĆ. 2000. Cross-serial dependencies in Dutch: Evidence for a discourse-based metric? Poster presented at CUNY Sentence Processing Conference, University of California, San Diego.
- KÖNIG, ESTHER. 1990. The complexity of parsing with extended categorial grammars. In *Proc. COLING 1990*.
- KRUIJFF, GEERT-JAN M. 1995. The Unbearable Demise of Surprise: Reflections on abduction in Artificial Intelligence and Peirce's philosophy. Master's thesis, University of Texas, UT.
- 1997. Peirce's theory of abduction and goedel's axioms of infinity - reflections on the mathematics of creativity. Presentation at LOGIKA'97, Liblice (Czech Republic). Available from <http://kwetal.ms.mff.cuni.cz/~gj/abduction.html>.
- 1998. Peirce's late theory of abduction: A comprehensive account. Available from <http://kwetal.ms.mff.cuni.cz/~gj/abduction.html>.
- 1999. Resource logics, grammars, and processing. In *Proceedings of the Workshop on Resource Logics and Minimalist Grammars*, ed. by Christian Retoré & Ed Stabler, Utrecht. ESSLLI.
- 2001. *A Categorial-Modal Architecture of Informativity: Dependency Grammar Logic and Information Structure*. Prague, Czech Republic: Charles University dissertation.
- LEWIS, RICHARD L. 1993. *An Architecturally-based theory of Human Sentence Comprehension*. Pittsburgh, PA: Carnegie Mellon University dissertation.
- 1998. Interference in working memory: Retroactive and proactive interference in parsing. Unpublished manuscript, The Ohio State University.
- MOORTGAT, MICHAEL. 1997. Categorial type logics. In *Handbook of Logic and Language*, ed. by Johan van Benthem & Alice ter Meulen, chapter 2. Amsterdam: Elsevier.
- NAKATANI, KENTARO, MARIA BABYONYSHEV, & EDWARD GIBSON. 2000. The complexity of nested structures in Japanese. Poster presented at the CUNY Sentence Processing Conference, University of California, San Diego.
- PARSONS, CHARLES. 1995. Platonism and mathematical intuition in Kurt Gödel's thought. *BSL* 1.44–74.
- SCHEEPERS, CHRISTOPH, BARBARA HEMFORTH, & LARS KONIECZNY. 1999. Incremental Processing of German Verb-Final Constructions: Predicting the Verb's Minimum (!) Valency. In *The Second International Conference on Cognitive Science, Japan*.

- SHIEBER, STUART. 1986. *Introduction to Unification-based approaches to grammar*. Stanford, CA: CSLI.
- SIKKEL, K. 1997. *Parsing Schemata*. Berlin: Springer.
- STRIGIN, A. 1998. Lexical rules as hypotheses generators. *Journal of Semantics* 15.163–190.
- UEHARA, KEIKO, & DIANNE BRADLEY. 1996. The effect of -ga sequences on processing Japanese multiply center-embedded sentences. In *11th Pacific-Asia Conference on Language, Information, and Computation*.
- VASISHTH, SHRAVAN. 2001. An empirical evaluation of sentence processing models: Center embeddings in Hindi. *Ohio State University Working Papers in Linguistics* 56.159–181.
- . in progress. A computational model of human sentence processing based on abductive inference. Master's thesis, Ohio State University, Columbus, OH.
- YAMASHITA, HIROKO. 1997. The Effects of Word Order and Case Marking Information on the Processing of Japanese. *Journal of Psycholinguistic Research* 26(2).163–188.